# Numerical Analysis.

## Matrix Algebra

notation : Let $n \in \mathbb{N}^*$. We denote $[n] = \{1, 2, \dots, n\}$.

Def : A matrix $M$ is a bijection $M : [m]\times[n] \to \mathbb{C}$. The set of all matrices is written $\mathbb{C}^{m\times n} = \{M \mid M : [m]\times[n] \to \mathbb{C}\}$.
A real matrix $M$ is a bijection $M : [m]\times[n] \to \mathbb{R}$.
and $\mathbb{R}^{m\times n} = \{M \mid M : [m]\times[n] \to \mathbb{R}\}$.

notation : $M(i,j) = m_{ij} = m[i,j]$. or :

$$M = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & \cdots & m_{mn} \end{bmatrix}$$

### ▼ Matrix Addition.

Def : Let $A, B \in \mathbb{C}^{m\times n}$.

$$C = A + B \iff \forall (i,j) \in [m]\times[n] : c_{ij} = a_{ij} + b_{ij}$$

Prop :

a) $A + B = B + A$ , $\forall A, B \in \mathbb{C}^{m\times n}$

b) $(A+B) + C = A + (B+C)$ , $\forall A, B, C \in \mathbb{C}^{m\times n}$.

### ▼ Matrix Multiplication

Def : Let $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times \ell}$ and $C \in \mathbb{C}^{m\times \ell}$

$$C = AB \iff c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \quad , \forall (i,j) \in [m]\times[\ell].$$

Prop:

a) $(AB)C = A(BC)$ , $\forall A \in \mathbb{C}^{m\times n}, B \in \mathbb{C}^{n\times k}, C \in \mathbb{C}^{k\times \ell}$

b) $A(B+C) = AB + AC$ , $\forall A \in \mathbb{C}^{m\times n}, B \in \mathbb{C}^{n\times \ell}, C \in \mathbb{C}^{n\times \ell}$.

c) $(B+C)A = BA + CA$ , $\forall B \in \mathbb{C}^{m\times n}, C \in \mathbb{C}^{m\times n}, A \in \mathbb{C}^{n\times \ell}$

## Scalar product

<u>Def</u>: Let $A \in \mathbb{C}^{m\times n}$ and $\lambda \in \mathbb{C}$.

$$B = \lambda A = A\lambda \Leftrightarrow b_{ij} = \lambda a_{ij} \quad , \forall (i,j) \in [n]\times[m].$$

<u>Prop</u>: For $\lambda \in \mathbb{C}$

a) $\lambda(A+B) = \lambda A + \lambda B \qquad , \forall A, B \in \mathbb{C}^{m\times n}$

b) $(\lambda+\mu)A = \lambda A + \mu A \qquad , \forall A \in \mathbb{C}^{m\times n}$

c) $(\lambda\mu)A = \lambda(\mu A) = \mu(\lambda A). \qquad , \forall A \in \mathbb{C}^{m\times n}$

d) $\lambda(AB) = (\lambda A)B = A(\lambda B). \qquad , \forall A, \in \mathbb{C}^{m\times n}, B \in \mathbb{C}^{n\times k}.$

## Transpose / Conjugate / Hermitians

<u>Def</u>: Let $A \in \mathbb{C}^{m\times n}$.

a) $B = A^T \in \mathbb{C}^{n\times m} \Leftrightarrow \forall (i,j) \in [m]\times[n] : a_{ij} = b_{ji}$ (transpose)

b) $B = A^* \in \mathbb{C}^{m\times n} \Leftrightarrow \forall (i,j) \in [m]\times[n] : b_{ij} = a^*_{ij}$ (conjugate)

c) $B = A^H \in \mathbb{C}^{n\times m} \Leftrightarrow \forall (i,j) \in [n]\times[m] : b_{ij} = a^*_{ji}.$ (hermitian)

<u>Prop</u>:

| | |
|---|---|
| a) $(A^T)^T = A$ | $(A^H)^H = A$ |
| b) $(\lambda A)^T = \lambda A^T$ | $(\lambda A)^H = \lambda A^H$ |
| c) $(A+B)^T = A^T + B^T$ | $(A+B)^H = A^H + B^H$ |
| d) $(AB)^T = B^T A^T$ | $(AB)^H = B^H A^H.$ |

## Special matrices.

<u>Def</u>:

a) The <u>zero matrix</u> : $A = \mathbf{0} \Leftrightarrow a_{ij} = 0 \quad , \forall (i,j) \in [m]\times[n].$

b) The <u>identity matrix</u> : $A = I \Leftrightarrow a_{ij} = \begin{cases} 1 & , i = j \\ 0 & , i \neq j. \end{cases}$

<u>Prop</u>

a) $A + \mathbf{0} = \mathbf{0} + A = A \qquad , \forall A$

b) $IA = AI = A. \qquad , \forall A$

**Def**: Let $A \in \mathbb{C}^{n \times n}$ (a square matrix).

1) $A$ symmetric $\Leftrightarrow A = A^T$
2) $A$ hermitian $\Leftrightarrow A = A^H$
3) $A$ orthogonal $\Leftrightarrow AA^T = A^T A = I$
4) $A$ unitary $\Leftrightarrow AA^H = A^H A = I$
5) $A$ diagonal $\Leftrightarrow a_{ij} = 0, \forall i \neq j$
6) $A$ upper triangular $\Leftrightarrow a_{ij} = 0, \forall i > j$
7) $A$ lower triangular $\Leftrightarrow a_{ij} = 0, \forall i < j$.

**Prop**: (Closure properties)

1) $A, B$ symmetric $\Rightarrow A+B, AB, \lambda A, A^T, A^H$ symmetric
2) $A, B$ hermitian $\Rightarrow A+B, AB, \lambda A, A^T, A^H$ hermitian
3) $A, B$ orthogonal $\Rightarrow AB, A^T$ orthogonal.
   - ~~If $A, B$ are real also then $A^T, B^T$ are orthogonal.~~
4) $A, B$ unitary $\Rightarrow AB, A^H, B^H$ unitary.
5) $A, B$ diagonal $\Rightarrow A+B, AB, \lambda A$ diagonal
6) $A, B$ upper triangular $\Rightarrow A+B, AB, \lambda A$ upper triangular and $A^T, A^H$ lower triangular.
7) $A, B$ lower triangular $\Rightarrow A+B, AB, \lambda A$ lower triangular and $A^T, A^H$ upper triangular.

## Matrix Inverse

**Def**: Let $A \in \mathbb{C}^{n \times n}$ be a square matrix.

$$B = A^{-1} \Leftrightarrow AB = BA = I.$$

**Prop**: (Inverse of orthogonal / unitary matrices).

a) If $A$ orthogonal $\Rightarrow A$ has inverse and $A^{-1} = A^T$.
b) If $A$ unitary $\Rightarrow A$ has inverse and $A^{-1} = A^H$.
c) If $A = \mathrm{diag}(a_1, a_2, \ldots, a_n)$ then
   $a_k \neq 0, \forall k \in [n] \Leftrightarrow A$ has inverse $A^{-1} = \mathrm{diag}(1/a_1, 1/a_2, \ldots, 1/a_n)$.

<u>Def</u> : (Permutations).

a) A bijection $\sigma : [n] \to [n]$ is called a permutation of order $n$.
   The set of all bijections $\sigma : [n] \to [n]$ is written as $S_n$.

b) Let $\sigma \in S_n$. The sign of $\sigma$ is:

$$\mathrm{sign}(\sigma) =$$

<u>Def</u> : (Determinants).

Let $A \in \mathbb{C}^{n \times n}$ be a matrix. The determinant of $A$ is:

$$\det A = \sum_{\sigma \in S_n} \mathrm{sign}(\sigma) \prod_{i \in [n]} a[i, \sigma(i)].$$

<u>Thm</u> : (Cramer's theorem).

a) Let $A \in \mathbb{C}^{n \times n}$. $A$ has a unique inverse $\Leftrightarrow \det A \neq 0$.

b) Let $A \in \mathbb{C}^{n \times n}$ with $\det A \neq 0$ and $x \in \mathbb{C}^{n \times 1}$ and $b \in \mathbb{C}^{n \times 1}$.
   If $A = [a_1 \, a_2 \cdots a_n]$ define
$$B_1 = [b \ a_2 \cdots a_n].$$
$$B_j = [a_1 \cdots a_{j-1} \ b \ a_{j+1} \cdots a_n].$$
$$B_n = [a_1 \cdots a_{n-1} \ b].$$
   Then, the solution to $Ax = b$ is unique and given by
$$x_j = \frac{\det B_j}{\det A}, \quad \forall j \in [n].$$

<u>Remark</u> : $A^{-1}$ can be computed this way by posing it as $n$ systems of the form $Ax = b$. Most numerical methods are based on the following theorem instead.

<u>Thm</u> : Let $A, B \in \mathbb{C}^{n \times n}$. Then ~~det(AB) = detA · detB.~~

a) $\det(AB) = \det A \cdot \det B$

b) $\det(A^T) = \det A$

c) $\det(\lambda A) = \lambda^n \det A$

## Vector spaces in $\mathbb{C}^n$.

<u>Def</u> : Let $S \subseteq \mathbb{C}^n$. We say that:

a) $S$ subspace of $\mathbb{C}^n \Leftrightarrow \forall v_1, v_2 \in S, \forall \lambda_1, \lambda_2 \in \mathbb{C} : \lambda_1 v_1 + \lambda_2 v_2 \in S$.

b) $S$ subspace of $\mathbb{R}^n \Leftrightarrow \forall v_1, v_2 \in S, \forall \lambda_1, \lambda_2 \in \mathbb{R} : \lambda_1 v_1 + \lambda_2 v_2 \in S$.

<u>Def</u> : Let $v_1, v_2, \dots, v_n \in \mathbb{C}^n$ be $n$ vectors. We say that

a) $v_1, v_2, \dots, v_n$ linearly indepedent $\Leftrightarrow$
$(\lambda_1 v_1 + \lambda_2 v_2 + \dots + \lambda_n v_n = \mathbf{0} \Rightarrow \lambda_1 = \lambda_2 = \dots = \lambda_n = 0)$.

b) $v_1, v_2, \dots, v_n$ linearly dependent $\Leftrightarrow v_1, \dots, v_n$ not linearly indepedent.

<u>Def</u> : Let $v_1, v_2, \dots, v_n \in \mathbb{C}^n$ be $n$ vectors. The space spanned by these vectors is defined as:

$$\operatorname{span}\{v_1, \dots, v_n\} = \left\{ \sum_{j=1}^{n} \lambda_j v_j \;\middle|\; \lambda_j \in \mathbb{C} \right\}.$$

<u>Prop</u> : Let $v_1, v_2, \dots, v_n \in \mathbb{C}^m$.

a) $\operatorname{span}\{v_1, v_2, \dots, v_n\}$ is a subspace of $\mathbb{C}^m$.

b) If $\{v_1, v_2, \dots, v_n\}$ are linearly indepedent, then
$b \in \operatorname{span}\{v_1, v_2, \dots, v_n\} \Leftrightarrow \exists$ unique $\lambda_1, \dots, \lambda_n \in \mathbb{C} : b = \lambda_1 v_1 + \lambda_2 v_2 + \dots + \lambda_n v_n$.

<u>Def</u> : Let $v_1, v_2, \dots, v_n \in \mathbb{C}^m$ and $S$ subspace of $\mathbb{C}^m$. We say that:

$$\{v_1, v_2, \dots, v_n\} \text{ basis of } S \Leftrightarrow \begin{cases} \{v_1, v_2, \dots, v_n\} \text{ linearly indepedent.} \\ S = \operatorname{span}\{v_1, v_2, \dots, v_n\} \end{cases}$$

<u>Thm</u> : Let $S$ be a subspace of $\mathbb{C}^m$.

a) $S$ has at least one basis set $B$.

b) If $B_1, B_2$ are basis to $S \Rightarrow |B_1| = |B_2|$.

<u>Remark</u>: The cardinality of basis sets to $S$ is a unique number which we call <u>dimension</u> of $S$. $\rightsquigarrow \dim S$.

c) If
$$\left.\begin{array}{l} B \subseteq S \\ |B| = \dim S. \\ B \text{ linearly indepedent} \end{array}\right\} \Rightarrow B \text{ a basis of } S.$$

## ▼ Standard vector spaces of a matrix.

Def : Let $A \in \mathbb{C}^{m\times n}$ be given.

a) $\text{range}(A) = \{y \in \mathbb{C}^m \mid \exists x \in \mathbb{C}^n : Ax = y\}$. (range of $A$).

b) $\text{null}(A) = \{x \in \mathbb{C}^n : Ax = 0\}$. (null space of $A$)

c) $\text{rank}(A) = \dim(\text{range}(A))$. (rank of $A$).

Prop. : Let $A = [a_1\, a_2 \cdots a_n] \in \mathbb{C}^{m\times n}$

a) $\text{span}\{a_1, a_2, \dots, a_n\} = \text{range}(A)$.

b) $\text{rank}(A) = \text{rank}(A^T)$.

c) $\dim(\text{null}(A)) + \text{rank}(A) = n$.

If in particular $A \in \mathbb{C}^{n\times n}$ is a square matrix:

d) $\text{rank}(A) = n \Leftrightarrow \text{null}(A) = \{0\} \Leftrightarrow \det A \neq 0 \Leftrightarrow \text{range}(A) = \mathbb{C}^n$.

## ▼ Orthogonality.

Def : Let $x, y \in \mathbb{C}^n$ be two vectors. We define their dot product by :

$$\langle x|y\rangle = \sum_{j=1}^{n} x_j^* y_j \in \mathbb{C}.$$

Def : Let $S = \{x_1, x_2, \dots, x_n\} \in \mathbb{C}^m$. We say that :

a) $S$ are orthogonal $\Leftrightarrow \forall x, y \in S : x \neq y \Rightarrow \langle x|y\rangle = 0$.

b) $S$ are orthonormal $\Leftrightarrow \begin{cases} \forall x \in S : \langle x|x\rangle = 1. \\ S \text{ orthogonal}. \end{cases}$

Prop. : Let $S = \{x_1, x_2, \dots, x_n\} \in \mathbb{C}^m$.

a) $S$ is orthogonal $\Rightarrow$ $S$ is linearly indepedent.

b) $A = [x_1 x_2 \cdots x_n]$ unitary $\Leftrightarrow$ $S$ orthonormal !!!

$A = [x_1\, x_2 \cdots x_n]$ orthogonal $\Leftrightarrow$ $S \subseteq \mathbb{R}^m \wedge S$ orthonormal.

Def : Let $S_1, S_2$ be subspaces of $\mathbb{C}^n$.

a) $S_1 \perp S_2 \Leftrightarrow \forall x \in S_1, \forall y \in S_2 : \{x, y\}$ orthogonal

b) $S^\perp = \{y \in \mathbb{C}^n : \langle x|y\rangle = 0, \forall x \in S\} \rightsquigarrow$ (orthogonal complement).

Prop. : Let $A \in \mathbb{C}^{m\times n}$.
Then $\underline{\text{range}(A)^{\perp} = \text{null}(A^H)}$.

Def : Let $S$ be a subspace of $\mathbb{C}^m$ and $B$ a basis of $S$.
$B$ orthonormal basis of $S \Leftrightarrow \begin{cases} B \text{ is orthonormal} \\ \text{span}\, B = S. \end{cases}$

Prop : If $B_1$ is an orthonormal basis of $S$ and $B_1 \cup B_2$ an orthonormal basis of $\mathbb{C}^n$, then
$$S^{\perp} = \text{span}\, B_2.$$

## Vector and matrix norms.

### Vector norms.

Def : A vector norm on $\mathbb{C}^n$ is a function $f: \mathbb{C}^n \to \mathbb{R}$ such that:
a) $f(x) \geq 0 \quad , \forall x \in \mathbb{C}^n$
b) $f(x+y) \leq f(x) + f(y) \quad , \forall x, y \in \mathbb{C}^n$
c) $f(\lambda x) = |\lambda| f(x) \quad , \forall \lambda \in \mathbb{C}, \forall x \in \mathbb{C}^n$.

Def : Let $x \in \mathbb{C}^n$. ~~The p-norm~~
a) The p-norm of $x$ is : $\|x\|_p = \left( \sum_{j=1}^{n} |x_j|^p \right)^{1/p}$

b) The $\infty$-norm of $x$ is : $\|x\|_\infty = \lim_{p \to +\infty} \|x\|_p = \max_{i \in [n]} |x_i|$

Prop : The functions $f_p(x) = \|x\|_p$ and $f_\infty(x) = \|x\|_\infty$ are vector norms.

Prop : (Holder inequality).
a) If $(1/p) + (1/q) = 1 \Rightarrow \forall x, y \in \mathbb{C}^n : |\langle x|y \rangle| \leq \|x\|_p \|y\|_q$.
b) Special case $p = q = 2 \rightsquigarrow$ Schwartz - Cauchy inequality.
$\forall x, y \in \mathbb{C}^n : |\langle x|y \rangle| \leq \|x\|_2 \|y\|_2$.

Thm: (norm equivalence)
$\forall a,b \in (0,+\infty)$, $\exists c_1, c_2 \in (0,+\infty)$: $\forall x \in \mathbb{C}^n$: $c_1\|x\|_a \leqslant \|x\|_b \leqslant c_2\|x\|_a$.
In particular: $\forall x \in \mathbb{C}^n$:
a) $\|x\|_2 \leqslant \|x\|_1 \leqslant \sqrt{n}\,\|x\|_2$.
b) $\|x\|_\infty \leqslant \|x\|_2 \leqslant \sqrt{n}\,\|x\|_\infty$
c) $\|x\|_\infty \leqslant \|x\|_1 \leqslant n\,\|x\|_\infty$.

## Matrix norms

Def: A matrix norm is a function $f: \mathbb{C}^{m\times n} \to \mathbb{R}$ such that
a) $f(A) \geqslant 0$, $\forall A \in \mathbb{C}^{m\times n}$
b) $f(A+B) \leqslant f(A)+f(B)$. , $\forall A,B \in \mathbb{C}^{m\times n}$
c) $f(\lambda A) = |\lambda| f(A)$ , $\forall \lambda \in \mathbb{C}$, $\forall A \in \mathbb{C}^{m\times n}$.

Def: Let $A \in \mathbb{C}^{m\times n}$.
a) The Frobenius norm of $A$ is given by:

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2}.$$

b) The p-norm of $A$ is given by $\|A\|_p = \sup_{x \in \mathbb{C}^n - \{0\}} \dfrac{\|Ax\|_p}{\|x\|_p}$.

Prop: The Frobenius norm and the p-norm of a matrix $A \in \mathbb{C}^{m\times n}$ are matrix norms.

Prop.
a) $\forall A \in \mathbb{C}^{m\times n}, \forall B \in \mathbb{C}^{n\times q}$: $\|AB\|_p \leqslant \|A\|_p \|B\|_p$.
b) $\forall A \in \mathbb{C}^{m\times n}, \forall x \in \mathbb{C}^n$: $\|Ax\|_p \leqslant \|A\|_p \|x\|_p$.
c) $\forall A \in \mathbb{C}^{m\times n}$: $\|A\|_1 = \max_{j\in[n]}\left[\sum_{i\in[m]} |a_{ij}|\right]$
d) $\forall A \in \mathbb{C}^{m\times n}$: $\|A\|_\infty = \max_{i\in[m]}\left[\sum_{j\in[n]} |a_{ij}|\right]$.
e) $\forall A \in \mathbb{C}^{m\times n}$: $\max_{(i,j)\in[m]\times[n]} |a_{ij}| \leqslant \|A\|_2 \leqslant \sqrt{mn}\max_{(i,j)\in[m]\times[n]} |a_{ij}|$.

f) $\forall A \in \mathbb{C}^{m\times n}: \|A\|_2 \leq \|A\|_F \leq \sqrt{n}\,\|A\|_2$.

## Singular value decomposition.

Thm : (SVD theorem).

$\forall A \in \mathbb{C}^{m\times n}, \exists \begin{cases} U = [u_1, \ldots, u_m] \in \mathbb{C}^{m\times m} \\ V = [v_1, \ldots v_n] \in \mathbb{C}^{n\times n} \end{cases}$ such that :

a) $U, V$ are unitary.
b) $U^* A V = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p) \in \mathbb{C}^{m\times n}$ , $p = \min\{m, n\}$.
c) $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.

nomenclature : $\sigma_i \rightsquigarrow$ singular values of $A$.
$u_i \rightsquigarrow$ $i^{th}$ left singular vector.
$v_i \rightsquigarrow$ $i^{th}$ right singular vector,

Thm : (Rank deficiency).
Let $A \in \mathbb{C}^{m\times n}$ have singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0$
Then : $\mathrm{rank}(A) = r$
$\mathrm{null}(A) = \mathrm{span}\{v_{r+1}, \ldots, v_n\}$
$\mathrm{range}(A) = \mathrm{span}\{u_1, \ldots, u_r\}$.

Thm : (Norms and SVD).
Let $A \in \mathbb{C}^{m\times n}$. have singular values $\sigma_1, \sigma_2, \ldots, \sigma_p$.
a) $\|A\|_2 = \sigma_1$.
b) $\|A\|_F = (\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2)^{1/2}$.

Def : Let $A \in \mathbb{C}^{m\times n}$ and $S = \{B \in \mathbb{C}^{m\times n} : \|A - B\|_2 \leq \varepsilon\}$. The $\varepsilon$-rank of $A$ is defined as

$$\mathrm{rank}(A, \varepsilon) = \min_{B \in S} \mathrm{rank}(B).$$

Remark : The concept of $\varepsilon$-rank helps characterize matrices that are nearly rank deficient. Of course:

$$\lim_{\varepsilon \to 0} \mathrm{rank}(A, \varepsilon) = \mathrm{rank}\, A.$$

<u>Thm</u>: Let $A \in \mathbb{C}^{m \times n}$ with singular values $\sigma_1, \sigma_2, \dots, \sigma_p$. and let $S_k = \{B \in \mathbb{C}^{m \times n} : \operatorname{rank} B = k\} \rightsquigarrow$ $m \times n$ matrices of rank $k$. with $k \in [p]$. Then

$$\min_{B \in S_k} \|A - B\|_2 = \sigma_{k+1}$$

<u>Remark</u>: $\sigma_{k+1}$ is the "distance" of $A$ from the "nearest" matrix $B$ with rank $k$.

<u>Thm</u>: Let $A \in \mathbb{C}^{m \times n}$ with singular values $\sigma_1, \sigma_2, \dots, \sigma_p$.
$\operatorname{rank}(A, \varepsilon) = r \Leftrightarrow \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \varepsilon \geq \sigma_{r+1} \geq \dots \geq \sigma_p$.
where $p = \min\{m, n\}$.

<u>Remark</u>: This theorem allows us to ~~classify~~ characterize the numerical rank of a matrix.

## ▼ <u>Condition number and SVD.</u>

<u>Def</u>: Let $A \in \mathbb{C}^{n \times n}$ with $\det A \neq 0$. We define the <u>condition number</u> by $k_p(A) = \|A\|_p \|A^{-1}\|_p$.

<u>Thm</u>: Let $A \in \mathbb{C}^{n \times n}$ with $\det A \neq 0$. and
$S = \{B \in \mathbb{C}^{n \times n} : \det B = 0\} \rightsquigarrow$ space of singular matrices.
Then

$$\min_{B \in S} \frac{\|A - B\|_p}{\|A\|_p} = \frac{1}{k_p(A)}$$

<u>Remark</u>: The condition number measures the "distance" of $A$ from the space of singular matrices. A matrix with large $k_p(A)$ is nearly singular $\rightsquigarrow$ ill-conditioned.

<u>Thm</u>: Let $A \in \mathbb{C}^{n \times n}$ with singular ~~singu~~ values $\sigma_1, \sigma_2, \dots, \sigma_n$. Then

$$k_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)}$$

<u>Remark</u>: SVD provides with a way to determine whether a matrix is ill-conditioned or not.

<u>Thm</u> (equivalence theorem)
$\forall a, b \in (0,+\infty), \exists c_1, c_2 \in (0,+\infty) : \forall A \in \mathbb{C}^{n\times n} : c_1 \kappa_a(A) \leq \kappa_b(A) \leq c_2 \kappa_a(A)$.
In particular, $\forall A \in \mathbb{C}^{n\times n}$:

$$\frac{1}{n} \kappa_2(A) \leq \kappa_1(A) \leq n \kappa_2(A).$$
$$\frac{1}{n} \kappa_\infty(A) \leq \kappa_2(A) \leq n \kappa_\infty(A).$$
$$\frac{1}{n^2} \kappa_1(A) \leq \kappa_\infty(A) \leq n^2 \kappa_1(A).$$

<u>Remark</u> : If a matrix is ill-conditioned, we will see that in $\kappa_a(A)$ indepedent of $a \in (0,+\infty)$.

<u>Prop.</u> : Let $A, B \in \mathbb{C}^{n\times n}$. Then $\kappa_p(AB) \leq \kappa_p(A)\kappa_p(B)$.

## Fundamental problems in linear algebra

1) Linear systems of equations.
   Let $A \in \mathbb{C}^{n\times n}$ and $b \in \mathbb{C}^n$ be given. Want $x \in \mathbb{C}^n$ such that $Ax = b$.
2) Eigenvalue problem.
   Let $A \in \mathbb{C}^{n\times n}$. Want $(\lambda, x) \in \mathbb{C} \times \mathbb{C}^n$ such that $Ax = \lambda x$.
3) Underdetermined system of equations.
   Want to solve $Ax = b$ for $A \in \mathbb{C}^{m\times n}$, $x, b \in \mathbb{C}^n$ with $m < n$.
4) Overdetermined system of equations.
   Same, for $m > n$.

## Linear systems of equations.

Cramer's theory provides an analytic closed-form solution.
Numerical cost is $n!$ $\rightsquigarrow$ not good. Use other methods.

### ▼ Special cases

Case 1: If $A \in \mathbb{C}^{n\times n}$ is unitary, then the following theorem applies:

Thm: If $A$ unitary $\Rightarrow A^{-1} = A^H$.
So $Ax = b \Leftrightarrow x = A^H b$.

Moreover, if $A = [a_1\ a_2 \cdots a_n]$, then $\det A = \prod_{j=1}^{n} \|a_j\|$.

Case 2: If $A \in \mathbb{C}^{n\times n}$ is lower triangular, then $\det A = \prod_{i=1}^{n} a_{ii}$ and the solution to $Ax = b$ can be computed recursively by:

$$\begin{cases} x_i = \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j \right) / a_{ii}, & i = 2, \dots, n \\ x_1 = b_1 / a_{11}. \end{cases}$$

Case 3 : If $A \in \mathbb{C}^{n\times n}$ is upper triangular, then $\det A = \prod_{i=1}^{n} a_{ii}$ and the solution to $Ax = b$ can be computed recursively by:

$$x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij}x_j\right)/a_{ii} \quad , i = n, \ldots, 2, 1.$$

$$x_n = b_n/a_{nn}.$$

Another special case: If it is "easy" to solve $Ax = b$, then it is also possible to solve $(A + UV^T)x = b$ where $U, V \in \mathbb{C}^{n\times k}$, using the following theorem:

Thm : (Sherman-Morrison-Woodbury formula).

a) If $\det(I + V^T A^{-1} U) \neq 0 \Rightarrow (A+UV^T)^{-1} = A^{-1} - A^{-1} U (I + V^T A^{-1} U)^{-1} V^T A^{-1}$.

b) If $Ay = b$ and $Az = u$, then $(A + uv^H)x = b$ has the solution:

$$x = y - \frac{\langle v|y\rangle}{1 + \langle v|z\rangle} z$$

If the SVD decomposition $A = U\Lambda V^H$ of $A$ is known, then $Ax = b$ has the solution $x = V\Lambda^{-1}U^H b$, where

$$\Lambda^{-1} = \mathrm{diag}(1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_n).$$

The SVD can also give tons of ~~better~~ more info.

other approaches:

1) QR decomposition.
2) LU decomposition.

are less expensive.

## QR decomposition

Thm : $\forall A \in \mathbb{C}^{m\times n}, \exists Q \in \mathbb{C}^{m\times n}, \exists R \in \mathbb{C}^{n\times n}$ such that

a) $Q$ is unitary.
b) $R$ is right triangular.
c) $A = QR$.

Proof: We will give an algorithm to compute $Q$ and $R$ □.

The general idea is to eliminate the lower triangular part of a matrix with successive <u>unitary</u> transformations:

$$A = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}, \quad H_1 A = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix}, \quad H_2 H_1 A = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix}.$$

Eventually $H_n H_{n-1} \cdots H_1 A = R \rightsquigarrow$ right triangular.

If $H_1, H_2, \ldots, H_n$ are unitary we can write

$$A = (H_1^H H_2^H \cdots H_n^H) R = QR.$$

There are two kinds of transform we can use.

1) Householder transforms.
2) Givens rotations.

## ▼ Householder reflections.

<u>Def</u> : Let $u \in \mathbb{C}^n$ and $v \in \mathbb{C}^n$ and define $A = [a_{ij}]$ by $a_{ij} = u_i^* v_j$

We write $A = u \otimes v = u^H v$.

<u>Def</u> : Let $u \in \mathbb{C}^n$. The Householder reflection around $u$ is defined:

$$H(u) = I - \frac{2}{\langle u | u \rangle} u \otimes u . \in \mathbb{C}^{n \times n}$$

<u>Thm</u> :

a) $H(u)$ is symmetric and unitary, $\forall u \in \mathbb{C}^n$.

b) Let $x \in \mathbb{C}^n$ and $z = (1, 0, 0, \ldots, 0) \in \mathbb{C}^n$. Then

$$H(x \pm \|x\|_2 z) x = \mp \|x\|_2 z.$$

<u>example</u>

For $x = (3, 1, 5, 1) \Rightarrow u = x + \|x\|_2 z = (9, 1, 5, 1) \Rightarrow$

$$\Rightarrow H(u) = \frac{1}{54} \begin{bmatrix} -27 & -9 & -45 & -9 \\ -9 & 53 & -5 & -1 \\ -45 & -5 & 29 & -5 \\ -9 & -1 & -5 & 53 \end{bmatrix}$$ has the property $H(u)x = (-6, 0, 0, 0)$.

Clever use of Householder transforms allows us to zero out the lower triangular part of A.

<u>Remark.</u>: There is a way to compute $H(u)x$ without forming $H(u)$:
Let $x, u \in \mathbb{C}^n$. Then:

$$y = H(u)x = \left(I - \frac{2}{\|u\|^2} uu^H\right)x = x - \frac{2}{\|u\|^2}(uu^H)x \Rightarrow$$

$$\Rightarrow y_i = x_i - \frac{2}{\|u\|^2}\sum_{j=1}^{n} u_i u_j^* x_j = x_i - \frac{2u_i}{\|u\|^2}\sum_{j=1}^{n} u_j^* x_j =$$

$$= x_i - \frac{2\langle u|x\rangle}{\langle u|u\rangle} u_i .$$

Therefore, the following algorithm will compute $H(u)x$.
<u>Algorithm</u>: Let $x, u \in \mathbb{C}^{n \times n}$.
$\lambda = -2\langle u|x\rangle / \langle u|u\rangle$.
loop $i = 1, \dots, n$: $y_i = x_i + \lambda u_i$.

Suppose that columns $j = 1, \dots, k-1$ of $A \in \mathbb{C}^{n \times n}$ are upper triangular.
To make column $j = k$ upper triangular apply $H(u)$, where
$u = x \pm \|x\| z$ and.

$$x = (\underbrace{0, 0, \dots, 0}_{k-1}, a_{kk}, a[k+1, k], \dots, a_{nk}).$$

$$z = (\underbrace{0, 0, \dots, 0}_{k-1}, 1, 0, \dots, 0)$$

To compute $H(u)A$ we apply $H(u)$ on columns $k, k+1, \dots, n$ of $A$.
If we do this for all $n-1$ columns we obtain:

$$H(u_{n-1}) \cdots H(u_2) H(u_1) A = R \Rightarrow$$

$$\Rightarrow A = \left(H(u_{n-1}) \cdots H(u_1)\right)^{-1} R =$$

$$= \left(H^H(u_1) H^H(u_2) \cdots H^H(u_{n-1})\right) R = QR.$$

## Givens rotations - real case

Let $s,c \in \mathbb{R}$ such that $s^2+c^2=1$ and $x_1,x_2 \in \mathbb{R}$.
Want $s,c$ such that $\begin{bmatrix} c & -s \\ s & c \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$

Solve: $\begin{cases} sx_1+cx_2=0. \\ s^2+c^2=1 \end{cases}$, $s=-c(x_2/x_1)$ therefore

$$s^2+c^2=1 \Leftrightarrow c^2\left(\frac{x_2^2}{x_1^2}+1\right)=1 \Leftrightarrow c^2\,\frac{x_1^2+x_2^2}{x_1^2}=1 \Leftrightarrow c^2=\frac{x_1^2}{x_1^2+x_2^2}$$

Choose $c=\dfrac{x_1}{\sqrt{x_1^2+x_2^2}}$. Then $s=-c(x_2/x_1)=\dfrac{-x_2}{\sqrt{x_1^2+x_2^2}}$.

Idea: We use transformations such as these to eliminate $A \sim$ Givens rotation

Def: Let $n \in \mathbb{N}^*$ and $\vartheta \in \mathbb{R}$ and $(i,k) \in [n]$. We define the Given's rotation $G_{ik}(\vartheta) \in \mathbb{R}^{n\times n}$ to be the matrix

$$G_{ik}(\vartheta) = I + (E_{ii}+E_{kk})\cos\vartheta + (E_{ki}-E_{ik})\sin\vartheta$$

where $E_{ij} \in \mathbb{R}^{n\times n}$ has all components equal to $0$ and the $(i,j)$ component equal to $1$.

Thm: $G_{ik}(\vartheta)$ is an orthogonal matrix with $G_{ik}^{-1}(\vartheta) = G_{ik}(-\vartheta)$.

To eliminate $A \in \mathbb{R}^{n\times n}$ we apply: $G_{n-1}G_{n-2}\cdots G_1 A = R$
where:

$$\begin{aligned}
G_1 &= G_{12}(\vartheta_{12})\,G_{13}(\vartheta_{13})\cdots G_{1n}(\vartheta_{1n}).\\
G_2 &= G_{23}(\vartheta_{23})\,G_{24}(\vartheta_{24})\cdots G_{2n}(\vartheta_{2n}).\\
&\;\vdots\\
G_{n-1} &= G_{n-1,n}(\vartheta_{n-1,n}).
\end{aligned}$$

where we choose the angles $\vartheta$ (actually $c,s$) such that they zero out the lower triangular part of $A$.

## Givens rotations - Complex case

Let $c \in \mathbb{R}, s \in \mathbb{C}$ such that $c^2+|s|^2=1$ and $z_1, z_2 \in \mathbb{C}$.
Want $c, s$ such that $\begin{bmatrix} c & s^* \\ -s & c \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} w \\ 0 \end{bmatrix}$.

Let $z_1 = x_1 + y_1 i$, $z_2 = x_2 + y_2 i$, $s = a + bi$.
$c^2 + |s|^2 = 1 \Leftrightarrow a^2 + b^2 + c^2 = 1$ (1).

Want $-sz_1 + cz_2 = 0 \Leftrightarrow sz_1 = cz_2 \Leftrightarrow (a+bi)(x_1+y_1 i) = c(x_2+y_2 i) \Leftrightarrow$
$\Leftrightarrow (ax_1 - by_1) + (ay_1 + bx_1)i = cx_2 + cy_2 i \Leftrightarrow$
$\Leftrightarrow \begin{cases} ax_1 - by_1 = cx_2 \\ ay_1 + bx_1 = cy_2 \end{cases} \Leftrightarrow \begin{bmatrix} x_1 & -y_1 \\ y_1 & x_1 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} cx_2 \\ cy_2 \end{bmatrix} \Leftrightarrow$
$\Leftrightarrow \begin{bmatrix} a \\ b \end{bmatrix} = \dfrac{1}{x_1^2+y_1^2}\begin{bmatrix} x_1 & y_1 \\ -y_1 & x_1 \end{bmatrix}\begin{bmatrix} cx_2 \\ cy_2 \end{bmatrix} \Leftrightarrow$

$$\Leftrightarrow \quad a = \frac{x_1x_2 + y_1y_2}{x_1^2+y_1^2}\, c$$
$$b = \frac{-x_2y_1 + x_1y_2}{x_1^2+y_1^2}\, c$$

Let $\lambda_1 = \dfrac{x_1x_2+y_1y_2}{x_1^2+y_1^2}$ and $\lambda_2 = \dfrac{x_1y_2 - x_2y_1}{x_1^2+y_1^2}$

Then $a = \lambda_1 c$ and $b = \lambda_2 c$ so
$a^2+b^2+c^2 = 1 \Leftrightarrow c^2(\lambda_1^2+\lambda_2^2+1) = 1 \Leftrightarrow c^2 = \dfrac{1}{\lambda_1^2+\lambda_2^2+1}$.

Choose $c = \dfrac{1}{\sqrt{\lambda_1^2+\lambda_2^2+1}}$.

Then $a = \dfrac{\lambda_1}{\sqrt{\lambda_1^2+\lambda_2^2+1}}$ and $b = \dfrac{\lambda_2}{\sqrt{\lambda_1^2+\lambda_2^2+1}}$.

The same idea as before

<u>Def</u> : Let $n \in \mathbb{N}^*$ and $c \in \mathbb{R}$, $s \in \mathbb{C}$. and $i,k \in [n]$. A Given's rotation $G_{ik}(s,c) \in \mathbb{C}^{n\times n}$ is the matrix:

$$G_{ik}(s,c) = I + (E_{ii}+E_{kk})c + E_{ik}s^* - E_{ki}s$$

<u>Thm</u> : If $|s|^2+c^2=1 \Rightarrow G_{ik}(s,c)$ is unitary. with $G^{-1}_{ik}(s,c) = G_{ik}(-s,c)$.

<u>Remark</u> : Given's rotations can be very flexible when working with banded matrices. They are also more stable than Householder with ill-conditioned matrices.

## ▼ Applying the QR decomposition

Let $A \in \mathbb{C}^{n\times n}$ with QR decomposition. $A = QR$.

① The linear system $Ax=b$ can be solved as follows:

$Ax = b \Leftrightarrow QRx = b \Leftrightarrow x = R^{-1}Q^T b$.

② The determinant can be computed by $\det A = \det(QR) = \det Q \cdot \det R$ and

$$\det R = \prod_{i=1}^{n} r_{ii}$$

$$\det Q = \prod_{j=1}^{n}\left(\sum_{i=1}^{n} q_{ij}^2\right)^{1/2}$$

The QR method is more robust. The LU method, to be discussed, is more efficient.

## LU decomposition

**Thm**: Let $A \in \mathbb{C}^{n\times n}$ be a square matrix and $A_K \in \mathbb{C}^{K\times K}$ a $K\times K$ submatrix of $A$ such that
$\forall (i,j) \in [K]\times[K] : A_K(i,j) = A(i,j)$
If $\forall K \in [n-1] : \det A_K \neq 0 \Rightarrow \exists\, L, U \in \mathbb{C}^{n\times n}$ such that:
a) $L$ is lower triangular
b) $U$ is upper triangular.
c) $l_{ii} = 1$, $\forall i \in [n]$.
d) $LU = A$.

If a matrix $A$ has an LU decomposition, then the $Ax = b$ problem can be solved easily since $L$ and $U$ are easy to invert.
If it exists, the decomposition has the form:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

This expands to the following $n^2$ equations:
$i<j$ : $l_{i1}u_{1j} + l_{i2}u_{2j} + \cdots + l_{ii}u_{ij} = a_{ij}$
$i=j$ : $l_{i1}u_{1j} + l_{i2}u_{2j} + \cdots + l_{ii}u_{jj} = a_{ij}$
$i>j$ : $l_{i1}u_{1j} + l_{i2}u_{2j} + \cdots + l_{ij}u_{jj} = a_{ij}$.
There are $n^2+n$ unknowns so we require $l_{ii} = 1$, $\forall i \in [n]$.
They can be solved recursively by Crout's algorithm:

**Algorithm**: (Crout's LU decomposition)
loop $i = 1, \ldots, n$ : $l_{ii} = 1$
loop $j = 1, 2, \ldots, n$
  loop $i = 1, \ldots, j$ : $u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}$

  loop $i = j+1, \ldots, n$ : $l_{ij} = \frac{1}{u_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}\right)$.

endloop □.
This algorithm merely rearranges the $n^2$ equations in the right order.

<u>Remark:</u> The existence theorem for the LU decomposition does not guarantee that all $A \in \mathbb{C}^{n \times n}$ can be decomposed. Some additional results follow:

<u>Def</u>: Let $A \in \mathbb{C}^{n \times n}$. We say that $A$ is <u>strictly diagonally dominant</u> iff

$$\forall i \in [n]: |a_{ii}| > \sum_{j \in [n] - \{i\}} |a_{ij}|$$

<u>Thm</u>: If $A^H$ is diagonally dominant $\Rightarrow$ $A$ has an LU decomposition.

$\hookrightarrow$ In this case the Crout algorithm will work.

<u>LU decomposition with pivoting.</u>

<u>Def</u>: Let $\sigma, \tau \in S_n$ be two permutations.
a) We define $\sigma\tau \in S_n$ by $(\sigma\tau)(i) = \sigma(\tau(i))$, $\forall i \in [n]$
b) and $j = \sigma^{-1}(i) \Leftrightarrow \sigma(j) = i$.

<u>Def</u>: Let $\sigma \in S_n$ and $A \in \mathbb{R}^{n \times n}$. We say that $A$ is the <u>permutation matrix</u> of $\sigma$ iff: $a_{ij} = \begin{cases} 1 & , j = \sigma(i) \\ 0 & , \text{otherwise}. \end{cases}$

<u>notation</u>: We write $A = P(\sigma)$.

<u>Prop</u>: Let $\sigma, \tau \in S_n$ be permutations.
a) $P(\sigma\tau) = P(\sigma)P(\tau)$.
b) $P(\sigma^{-1}) = P^{-1}(\sigma) = P^T(\sigma)$.
c) $\det P(\sigma) = \operatorname{sign}(\sigma)$.
d) $\operatorname{sign}(\sigma\tau) = \operatorname{sign}(\sigma)\operatorname{sign}(\tau)$.

For our analysis we are interested in a particular type of permutations.

**Def**: A permutation $\sigma \in S_n$ is a transposition $\Leftrightarrow$
$\Leftrightarrow \exists\, i,j \in [n] : \begin{cases} \sigma(i) = j \wedge \sigma(j) = i \\ \sigma(k) = k,\ \forall k \in [n] - \{i,j\}. \end{cases}$

notation: $\sigma = (i\ j)$.

**Prop**: If $\sigma \in S_n$ transposition $\Rightarrow \det P(\sigma) = \text{sign}(\sigma) = -1$

The importance of these facts follows from the following result:

**Thm**:
$\forall A \in \mathbb{C}^{n \times n}, \exists\, \sigma \in S_n : P(\sigma)A$ has an LU decomposition.

$\hookrightarrow$ The brute force approach is to try all permutations $\sigma \in S_n$ and find one for which the Crout algorithm works.
Better approach: Consider for a given $j \in [n]$ the equations:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \qquad i \in [j].$$

$$l_{ij} = \frac{1}{u_{jj}} \left( a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right). \qquad i \in [n] - [j].$$

Note that for $i=j$ the equation for $u_{jj}$ is identical to the equation for $l_{jj}$ except for the division.
Then we may permute row $j$ with a row $k > j$:

$$\begin{array}{c} \quad\ \ j \\ \begin{bmatrix} \times & \times & \times & \times \\ \times & \otimes & \times & \times \\ \times & \boxed{\times} & \times & \times \\ \times & \boxed{\times} & \times & \times \end{bmatrix} \end{array}$$

This effectively changes which matrix $A$ we decompose $\rightsquigarrow PA$.
It also makes sure that $u_{jj} \neq 0$ and $u_{jj} \gg 0$ so that the algorithm is stable.
From $u_{jj}, u_{j+1,j}, u_{j+2,j}, \ldots, u_{nj}$ we choose the largest $|u_{kj}|$ and swap rows. This way we avoid Crout's algorithm breaking down and we accumulate the row permutation $\sigma \in S_n$

that works. The algorithm now can be cast as follows:

**Algorithm** : (Crout with pivoting).

$\sigma = 1.$

\# loop $i = 1, \dots, n$ : $l_{ii} = 1$

loop $j = 1, 2, \dots, n$

  loop $i = 1, \dots, j$ : $u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$

  loop $i = j+1, \dots, n$ : $l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}$

  Select $l_{kj} = \max\{u_{jj}, l_{j+1,j}, l_{j+2,j}, \dots, l_{nj}\}$

  Swap rows $(j,k)$ of $a$.

  loop $i = j+1, \dots, n$ : $l_{ij} = l_{ij}/u_{jj}$.

  $\sigma = (j\ k)\sigma$  \# remember the row permutations so far.

endloop.

note : This algorithm works if $a_{ij}, l_{ij}, u_{ij}$ share the same memory block. At the end, the LU decomposition is stored in the following form:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{1n} \\ l_{21} & u_{22} & u_{23} & u_{2n} \\ l_{31} & l_{32} & u_{33} & u_{3n} \\ l_{n1} & l_{n2} & l_{n3} & u_{nn} \end{bmatrix}$$

It is implied that $l_{ii} = 1$ so no storage is necessary for that. The $\sigma$ accumulates $P(\sigma)$ such that

$$P(\sigma)A = LU.$$

**Applications**

① Let $A \in \mathbb{C}^{n\times n}$ and $b \in \mathbb{C}^n$ be given. Want $x \in \mathbb{C}^n$ such that $Ax = b$.

$Ax = b \iff P(\sigma)Ax = P(\sigma)b \iff LUx = P(\sigma)b \iff \underline{x = U^{-1}L^{-1}P(\sigma)b.}$

The $L^{-1}, U^{-1}$ can be applied with the help of the recursions we mentioned earlier. They can be done with the following algorithm:

<u>Algorithm</u> (back substitution).

$b = \sigma b$

$x_1 = b_1$

loop $i = 2, 3, \dots, n$ : $x_i = b_i - \sum_{j=1}^{i-1} a_{ij} x_j$

$x_n = x_n / a_{nn}$

loop $i = n-1, n-2, \dots, 1$ : $x_i = \frac{1}{a_{ii}} \left[ x_i - \sum_{j=i+1}^{n} a_{ij} x_j \right]$

↳ In this algorithm we assume that $l_{ii} = 1$ and that $a_{ij}$ stores both $L$ and $U$ as explained.

② <u>Determinants.</u>

$P(\sigma) A = LU \Rightarrow \det(P(\sigma)A) = \det(LU) \Rightarrow \det(P(\sigma)) \det A = \det(L) \det(U)$

$\Rightarrow \det A = \frac{\det L \cdot \det U}{\det(P(\sigma))} = \frac{\det L \cdot \det U}{\operatorname{sign}(\sigma)} = \operatorname{sign}(\sigma) \det L \cdot \det U$

Note that

$\det L = \prod_{i=1}^{n} l_{ii} = 1$ and $\det U = \prod_{i=1}^{n} u_{ii}$.

Finally notice that the LU decomp. algorithm forms $\sigma$ as a product of $d$ transpositions:

$$\sigma = \sigma_1 \sigma_2 \cdots \sigma_d$$

so by the established results:

$\operatorname{sign}(\sigma) = \prod_{i=1}^{n} \operatorname{sign}(\sigma_i) = (-1)^d$

It follows that $\det A = (-1)^d \prod_{i=1}^{n} u_{ii}$.

## Special linear systems problems.

### Tridiagonal systems.

Def : A matrix $A \in \mathbb{C}^{n \times n}$ is called tridiagonal iff
$a_{ij} \neq 0 \Leftrightarrow i=j \vee i=j+1 \vee i=j-1$.
If $A$ is tridiagonal, then the problem $Ax=b$ is called a tridiagonal system of equations.

Remark : A tridiagonal matrix can be written as:

$$A = \begin{bmatrix} d_0 & u_0 & 0 & 0 \\ l_1 & d_1 & u_1 & 0 \\ 0 & l_{n-2} & d_{n-2} & u_{n-2} \\ 0 & 0 & l_{n-1} & d_{n-1} \end{bmatrix}$$

In general, it is best to solve these problems with Givens rotations and QR decomposition. In many special cases however, the Thomas algorithm can be used instead.

Def : Let $A \in \mathbb{C}^{n \times n}$ be a tridiagonal matrix. We say that $A$ is diagonally dominant $\Leftrightarrow \forall i \in \{0, \dots, n-1\} : |d_i| > |l_i| + |u_i|$ where we assume $l_0 = 0$ and $u_{n-1} = 0$.

Thm : If $A \in \mathbb{C}^{n \times n}$ is a tridiagonal matrix which is diagonally dominant then it can be decomposed in LU:

$$A = \begin{bmatrix} d_0 & u_0 & 0 & 0 \\ l_1 & d_1 & u_1 & 0 \\ 0 & l_{n-2} & d_{n-2} & u_{n-2} \\ 0 & 0 & l_{n-1} & d_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_1 & 1 & 0 & 0 \\ 0 & a_{n-2} & 1 & 0 \\ 0 & 0 & a_{n-1} & 1 \end{bmatrix} \begin{bmatrix} b_0 & c_0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 \\ 0 & 0 & b_{n-2} & c_{n-2} \\ 0 & 0 & 0 & b_{n-1} \end{bmatrix}$$

by the Thomas algorithm:

<u>Algorithm</u> : (Thomas) $\rightsquigarrow$ specialization of Crout's algorithm (no pivotting).

$b_0 = d_0$
$a_1 = l_1 / d_0$
loop $k = 1, \dots, n-1$ :
    $a_k = l_k / b_{k-1}$
    $b_k = d_k - ul_{k-1} a_k$.
endloop
loop $k = 0, \dots, n-2$ : $c_k = u_k$.

A linear system $Ax = b$ can now be solved as follows:

<u>Algorithm</u>
$x = b$.
loop $k = 1, \dots, n-1$ : $x_k = x_k - a_k x_{k-1}$
$x_{n-1} = x_{n-1} / b_{n-1}$
loop $k = n-2, \dots, 1$ : $x_k = (x_k - c_k x_{k+1}) / b_k$.

## Rank-1 update of easy matrix.

**Def**: Let $u, v \in \mathbb{C}^n$ be two rectors. The <u>direct product</u> $A = u \otimes v \in \mathbb{C}^{n \times n}$ is defined by $a_{ij} = u_i v_j$.

**Def**: Let $A \in \mathbb{C}^{n \times n}$ and $u, v \in \mathbb{C}^n$. We say that $B = A + u \otimes v$ is a <u>rank-1 update</u> of $A$.

~~exaple~~

<u>examples</u>

1) Let $u_i = \begin{cases} 0 & , i \neq i_0 \\ 1 & , i = i_0 \end{cases}$ and $v_j = \begin{cases} 0 & , j \neq j_0 \\ \lambda & , j = j_0. \end{cases}$

Then $A + u \otimes v$ is thae matrix $A$ with $\lambda$ added to the $(i_0, j_0)$ entry.

2) Let $u_i = \begin{cases} 0 & , i \neq i_0 \\ 1 & , i = i_0 \end{cases}$ and $v_j = b_j - a_{i_0 j}$

The $A + u \otimes v$ is the matrix $A$ with the $i_0$ row replaced by the vector $b_j$.

3) Let $u_i = b_i - a_{i j_0}$ and $v_j = \begin{cases} 0 & , j \neq j_0 \\ 1 & , j = j_0 \end{cases}$

The $A + u \otimes v$ is the matrix $A$ with the $j_0$ column replaced by the vector $b_j$.

It follows that the following modifications are rank-1 updates:

a) Replacing a specific matrix element.
b) Replacing a column
c) Replacing a row.

**Def**: Let $A \in \mathbb{C}^{n \times n}$ and $u_1, u_2, \ldots, u_p, v_1, v_2, \ldots, v_p \in \mathbb{C}^n$. We say that:

$$B = A + \sum_{i=1}^{p} u_i \otimes v_i$$

is a <u>rank-2 update</u> of $A$.

If $A^{-1}$ is known or easy to compute (e.g. $A$ is tridiagonal) then the inverse of rank-1 and rank-2 updates of $A$ can be easily computed.

Algorithm : (Shermann - Morrisson)

Let $A \in \mathbb{C}^{n \times n}$ and $u, v \in \mathbb{C}^n$. and $b \in \mathbb{C}^n$ be given.
To solve $(A + u \otimes v)x = b$ :
Solve $Ay = b \rightsquigarrow y$.
Solve $Az = u \rightsquigarrow z$
$$x = y - \frac{\langle v | y \rangle}{1 + \langle v | z \rangle} z.$$

Algorithm : (Woodbury).

Let $A \in \mathbb{C}^{n \times n}$ and $u_i, v_i \in \mathbb{C}^n, \forall i \in [p]$ and $b \in \mathbb{C}^n$ be given.
The following algorithm solves :
$$\left(A + \sum_{i=1}^{p} u_i \otimes v_i\right)x = b.$$

loop $k = 1, \dots, p$ : Solve $Az_k = u_k \rightsquigarrow z_k$.
$Z = [z_1 \, z_2 \cdots z_p] \in \mathbb{C}^{n \times p}$
$V = [v_1 \, v_2 \cdots v_p]$
$H = (I + V^H Z). \in \mathbb{C}^{p \times p} \rightsquigarrow$ Capacitance matrix
Compute $H^{-1} \rightsquigarrow$ problem size is $p \times p$. Small $p \rightsquigarrow$ easy.
Solve $Ay = b \rightsquigarrow y$.
$x = y - Z \cdot [H^{-1} \cdot (V^H y)]$

## ▼ Positive definite A – Cholesky decomposition

<u>Def</u>: Let $x, y \in \mathbb{C}^n$ and $A \in \mathbb{C}^{n\times n}$. The <u>quadratic form</u> $\langle x|A|y\rangle$ is defined by:

$$\langle x|A|y\rangle = \langle x|Ay\rangle.$$

<u>Def</u>: Let $A\in\cancel{\mathbb{C}}\,\mathbb{R}^{n\times n}$ be a square matrix (real).

a) A <u>positive semidefinite</u> $\Leftrightarrow \forall x\in\mathbb{R}^n : \langle x|A|x\rangle \geqslant 0$

b) A <u>positive definite</u> $\Leftrightarrow \forall x\in\mathbb{R}^n : \langle x|A|x\rangle > 0$.

<u>Thm</u>: Let $A\in\mathbb{R}^{n\times n}$ be a matrix $A=[a_{ij}]$ and let $A_K = [a_{ij}] \in \mathbb{R}^{K\times K}$ be submatrices.

A positive definite $\Leftrightarrow \forall K\in[n] : \det A_K > 0$.

<u>Corollary</u>: If A positive definite $\Rightarrow$ A has an LU decomposition.

~~Remark: If $A\in\mathbb{R}^{n\times n}$ is positive definite~~ then the LU decomposition ~~is of the~~

# The eigenvalue problem

## Basic definitions.

<u>Def</u>: Let $A \in \mathbb{C}^{n\times n}$ be a complex square matrix.

a) The polynomial $p(z) = \det(zI - A) = c_n z^n + c_{n-1} z^{n-1} + \dots + c_0$ is called the <u>characteristic polynomial</u> of $A$.

b) The set $\lambda(A) = \{z \in \mathbb{C} : p(z) = 0\} = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ is called the <u>spectrum</u> of $A$.

c) We say that $\lambda \in \mathbb{C}$ <u>eigenvalue</u> of $A \Leftrightarrow \lambda \in \lambda(A)$.

<u>Def</u>: Let $A \in \mathbb{C}^{n\times n}$ and $\lambda \in \lambda(A)$.

a) The vector space $E(\lambda; A) = \{x \in \mathbb{C}^n : Ax = \lambda x\}$ is called the <u>eigenspace</u> of $\lambda$.

b) $x \in \mathbb{C}^n$ is an <u>eigenvector</u> of $A$ with eigenvalue $\lambda \Leftrightarrow x \in E(\lambda; A)$.

<u>Def</u>: Let $A, B \in \mathbb{C}^n$. We say that

$A, B$ are <u>similar</u> $\Leftrightarrow \lambda(A) = \lambda(B)$.

<u>Prop</u>: Let $A = \mathrm{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n) \in \mathbb{C}^{n\times n}$. Then

$\lambda(A) = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$.

<u>Remark</u>: ~~The purpos~~ To find $\lambda(A)$ we may either

a) Solve the equation $\det(zI - A) = 0$

b) Or find a diagonal matrix $D$ such that $\lambda(D) = \lambda(A)$

Numerically, (b) is the best approach.

## Similarity and invariance

<u>Def</u>: Let $S \subseteq \mathbb{C}^n$ be a subspace of $S$. We say that

$S$ is <u>invariant</u> wrt $A \Leftrightarrow \{Ax \mid x \in S\} = S$.

<u>Prop</u>: The eigenspaces $E(\lambda; A)$ for $\lambda \in \lambda(A)$ are invariant wrt to $A$.

**Def**: Let $S_1, S_2, \ldots, S_k$ be subspaces of $\mathbb{C}^n$. such that
$$S_i \cap S_j = \emptyset \quad , \forall (i,j) \in [k]\times[k] : i \neq j.$$
and let $B_1, B_2, \ldots, B_n$ be the corresponding basis:
$$S_i = \operatorname{span} B_i \quad , \forall i \in [k].$$
We define the ~~diret~~ <u>direct sum</u> $S$ of $S_1, S_2, \ldots, S_k$ by:
$$S = \bigoplus_{i=1}^{k} S_i = S_1 \oplus S_2 \oplus \cdots \oplus S_k = \operatorname{span}\left( \bigcup_{i=1}^{k} B_i \right).$$

**Remark**: The dimension of $S$ is
$$\dim S = \left| \bigcup_{i=1}^{k} B_i \right| = \sum_{i=1}^{k} |B_i| = \sum_{i=1}^{k} \dim S_k.$$

**Thm**: Let $S \subseteq \mathbb{C}^n$ be a subspace of $\mathbb{C}^n$ and $A \in \mathbb{C}^{n\times n}$.
If $S$ is invariant wrt to $A$, then
$$\exists \Lambda \subseteq \lambda(A) : \quad S = \bigoplus_{\lambda \in \Lambda} E(\lambda, A).$$

I.e: all invariant wrt $A$ subspaces of $\mathbb{C}^n$ are direct sums of some eigenspaces of $A$.

**Thm**: Let $A, B, X \in \mathbb{C}^{n\times n}$.

a) $AX = XB \Rightarrow \operatorname{range}(X)$ is invariant wrt $A$.

b) $\left.\begin{array}{l} AX = XB \\ \operatorname{rank} X = n \end{array}\right\} \Rightarrow \lambda(B) \subseteq \lambda(A)$. and

c) $\left.\begin{array}{l} AX = XB \\ \det X \neq 0 \end{array}\right\} \Rightarrow \lambda(B) = \lambda(A). \Rightarrow A, B$ are similar.
and $\operatorname{range} X$ is invariant wrt. $A$.

**Remark**: In case (c) $AX = XB \Rightarrow B = X^{-1}AX \rightsquigarrow$ <u>similarity transform.</u>

**Corollary**: Let $A \in \mathbb{C}^{n\times n}$ and $X \in \mathbb{C}^{n\times n}$ with $\det X \neq 0$.
Then $A$ and $X^{-1}AX$ are similar.

## Eigenvalue information.

**Def:** Let $A \in \mathbb{C}^{n\times n}$. The trace of $A$ is: $\operatorname{tr} A = \sum_{i=1}^{n} a_{ii}$.

**Thm:** Let $A \in \mathbb{C}^{n\times n}$ with spectrum $\Lambda(A)$.

a) The trace is the sum of the eigenvalues:

$$\operatorname{tr} A = \sum_{\lambda \in \Lambda(A)} \lambda.$$

b) The determinant is the product of the eigenvalues:

$$\det A = \prod_{\lambda \in \Lambda(A)} \lambda.$$

c) The condition number $\kappa_2(A)$ is given by:

$$\kappa_2(A) = \frac{\max\{|\lambda| : \lambda \in \lambda(A)\}}{\min\{|\lambda| : \lambda \in \lambda(A)\}}.$$

d) The singular values $\{\sigma_1, \sigma_2, \dots, \sigma_n\}$ of $A$ are the eigenvalues of $A^H A$: $\lambda(A^H A) = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$.

~~**Def:** Let $x, y \in \mathbb{C}^n$ and $A \in \mathbb{C}^{n\times n}$. $x, y \in$~~
~~a) The quadratic form: $\langle x|A|y\rangle = \langle x|Ay\rangle$.~~
~~b) $A$ is positive definite $\Leftrightarrow$~~

**Def:** Let $x, y \in \mathbb{C}^n$ and $A \in \mathbb{C}^{n\times n}$. The quadratic form $\langle x|A|y\rangle$ is defined by $\langle x|A|y\rangle = \langle x|Ay\rangle$.

**Def:** Let $A \in \mathbb{R}^{n\times n}$. We say that $A$ is positive definite $\Leftrightarrow$
$\Leftrightarrow \forall x \in \mathbb{C}^n : \langle x|A|x\rangle > 0$.
We say that $A$ positive semidefinite $\Leftrightarrow$
$\Leftrightarrow \forall x \in \mathbb{C}^n : \langle x|A|x\rangle \geq 0$.

**Thm**: Let $A \in \mathbb{C}^{n\times n}$.

a) $A$ positive definite $\Leftrightarrow \forall \lambda \in \lambda(A): \lambda > 0$

b) $A$ positive semidefinite $\Leftrightarrow \forall \lambda \in \lambda(A): \lambda \geq 0$.

**Def**: Let $A \in \mathbb{C}^{n\times n}$ and $f: \mathbb{C}^{n\times n} \to \mathbb{R}$ be a matrix norm.
We say that $A$ is <u>stable</u> $\Leftrightarrow$ The sequance $x_n = f(A^n)$ is bounded.

**Thm**: Let $A \in \mathbb{C}^{n\times n}$.

a) If $\max\{|\lambda| : \lambda \in \lambda(A)\} < 1 \Rightarrow A$ is stable.

b) If $\begin{cases} \max\{|\lambda| : \lambda \in \lambda(A)\} \leq 1. \\ |\lambda_1| = |\lambda_2| = 1 \Rightarrow \lambda_1 \neq \lambda_2, \forall \lambda_1, \lambda_2 \in \lambda(A) \end{cases} \Longleftrightarrow A$ is stable

**Remark/Def**: The quantity $\rho(A) = \max\{|\lambda| : \lambda \in A\}$ we call <u>spectral radius</u> of $A$.

## ▼ Schur decomposition.

**Thm**: (Schur theorem).
Let $A \in \mathbb{C}^{n\times n} \Rightarrow \exists\, Q \in \mathbb{C}^{n\times n}$ such that

a) $Q$ unitary.

b) $Q^H A Q = T$ upper triangular.

c) $\operatorname{diag} T = (\lambda_1, \lambda_2, \dots, \lambda_n)$ where $\lambda_1, \lambda_2, \dots, \lambda_n$ are eigenvalues of $A$.

**Remark**: If by applying unitary transforms $Q_1^H A Q_1$ we bring our matrix to upper triangular form then we have found the eigenvalues. What about eigenvectors?

**Def**: Let $Q^H A Q = T$ be the Schur decomposition of $A \in \mathbb{C}^{n\times n}$ and $Q = [q_1\, q_2 \cdots q_n]$.
The vectors $q_1, q_2, \dots, q_n \in \mathbb{C}^n$ are called the <u>Schur vectors</u> of $A$.

**Def**: Let $A \in \mathbb{C}^{n\times n}$. $A$ is normal $\Leftrightarrow A^H A = AA^H$.

**Thm**: Let $A \in \mathbb{C}^{n\times n}$.
a) If $A \in \mathbb{C}^{n\times n}$ is normal $\Rightarrow$ The Schur vectors are eigenvectors of $A$.
b) $\exists$ unitary $Q \in \mathbb{C}^{n\times n}$: $Q^H A Q = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$.

## Diagonalizibility.

**Def**: Let $A \in \mathbb{C}^{n\times n}$.
a) $A$ is diagonalizable $\Leftrightarrow \exists X \in \mathbb{C}^{n\times n}$: $X^{-1}AX$ is diagonal.
b) $A$ is defective $\Leftrightarrow A$ is not diagonalizable

**Thm**: If $A \in \mathbb{C}^{n\times n}$ is a diagonalizable matrix with
$X^{-1}AX = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$
and $X = [x_1\, x_2 \dots x_n]$ then $x_k \in E(\lambda_k, A)$.

**Remark**: It is nice when $A$ is diagonalizable. Most matrices of interest can be diagonalized Otherwize, the best we can do is:

**Thm**: (Jordan decomposition).
Let $A \in \mathbb{C}^{n\times n}$ with $t$ distinct eigenvalues $\lambda(A) = \{\lambda_1, \lambda_2, \dots, \lambda_t\}$.
Then $\exists X \in \mathbb{C}^{n\times n}$ : $X^{-1}AX = \mathrm{diag}(J_1, J_2, \dots, J_t)$ where

$$J_i = \begin{bmatrix} \lambda_i & 1 & \cdots & 0 \\ 0 & \lambda_i & \cdots & \vdots \\ \vdots & & \ddots & 1 \\ 0 & 0 & \cdots & \lambda_i \end{bmatrix} \in \mathbb{C}^{m(\lambda_i)\times m(\lambda_i)}.$$

and $\sum_{i=1}^{t} m(\lambda_i) = n$.

**Def**: The integers $m(\lambda_i)$ corresponding to eigenvalues of $A$ are called algebraic multiplicity of $\lambda_i$.

The following proposition gives an ~~eas~~ easy way to compute $m(\lambda)$:

<u>Prop.</u> : Let $A \in \mathbb{C}^{n \times n}$ with characteristic polynomial $p(z)$.
$m(\lambda) =$ root multiplicity of $p(z)$. root $\lambda$.

<u>Def</u> : Let $A \in \mathbb{C}^{n\times n}$ and $\lambda \in \lambda(A)$.
geometric multiplicity of $\lambda = \dim E(\lambda_0; A)$

<u>Remark.</u>: The geometric multiplicity of an eigenvalue can be computed as follows:

$$
\begin{aligned}
\dim E(\lambda, A) &= \dim \{x \in \mathbb{C}^n : Ax = \lambda x\} = \\
&= \dim \{x \in \mathbb{C}^n : (A - \lambda I)x = 0\} = \\
&= \dim (null(A - \lambda I)) = \quad \rightsquigarrow (\dim(null A) + rank A = n.) \\
&= n - rank(A - \lambda I). \quad \square.
\end{aligned}
$$

Diagonalizibility is determined by the following result:

<u>Thm</u> : Let $A \in \mathbb{C}^{n\times n}$.
$A$ diagonalizable $\Leftrightarrow \dim E(\lambda, A) = m(\lambda) \quad, \forall \lambda \in \lambda(A)$.

<u>Remark</u>
If $A \in \mathbb{C}^{n\times n}$ is defective then an $X \in \mathbb{C}^{n\times n}$ and a diagonal matrix $D$ still exist such that $AX = XD$. However $X$ has no inverse and not all of it's columns are eigenvectors.

<u>Thm</u> : Let $A \in \mathbb{C}^{n\times n}$. Define recursively:
$Q_0 R_0$ is QR-decomposition of $A$
~~$A_{K+1} = R_K Q_K$~~.
$Q_K R_K$ is QR-decomposition of $A_K$.
$A_{K+1} = R_K Q_K$.
If $A$ is diagonalizable $\Rightarrow \lim_{K \to +\infty} A_K$ is upper triangular.

<u>Remark</u>: Numerical methods used to solve the eigenvalue problem are based on this theorem and the assumption

that $A$ is diagonalizable.
If a matrix is diagonalizable, we can determine whether it is stable:

<u>Def</u> : Let $A \in \mathbb{C}^{n \times n}$ and $f : \mathbb{C}^{n \times n} \to \mathbb{R}$ a matrix norm.
We say that $A$ <u>stable</u> $\Leftrightarrow$ The sequence $x_n = f(A^n)$ is bounded.

<u>Thm</u> : Let $A \in \mathbb{C}^{n \times n}$ be a <u>diagonalizable</u> matrix

~~$A$ stable $\Longleftrightarrow \begin{cases} \max\{|\lambda| : \lambda \in \lambda(A)\} \leq 1 \\ \forall \lambda_1, \lambda_2 \in \lambda(A) : |\lambda_1| = |\lambda_2| = 1 \Rightarrow \end{cases}$~~

$A$ stable $\Longleftrightarrow \begin{cases} \max\{|\lambda| : \lambda \in \lambda(A)\} \leq 1 \\ \forall \lambda \in \lambda(A) : |\lambda| = 1 \Rightarrow m(\lambda) = 1. \end{cases}$

<u>Remark</u> : The quantity $\rho(A) = \max\{|\lambda| : \lambda \in A\}$ is called the <u>spectral radius.</u> of $A$.

## Eigenvalue problem numerical methods.

To find the eigenvalues of a matrix $\mathbb{C}^{n\times n}$.
1) Balance the matrix
2) Reduce Hessenberg form.
3) Compute the Schur decomposition. $\rightsquigarrow$ eigenvalues.

If the matrix is symmetric, the Schur decomposition also gives the eigenvectors. General case:
a) Use SVD decomposition ~~of~~ on $A-\lambda I$ for $\lambda \in \lambda(A)$.
b) The characteristic polynomial method
c) Inverse iteration.

(a) most robust but inefficient
(b),(c) good methods.

main idea: Use transformations of the form $A \longrightarrow X^{-1}AX$ to bring the matrix to Schur form. Best to use unitary $X$, otherwise condition number will increase.

## Matrix balancing.

It is necessary for non-symmetric matrices. $\rightsquigarrow$ reduces eigenvalue sensitivity.

Transform $B = D^{-1}AD$ where $D$ is a diagonal matrix such that $B$ rows and columns have the same infinity norm. That is if:

$$B = \mathrm{col}(c_1, c_2, \ldots, c_n) = \mathrm{row}(r_1, r_2, \ldots, r_n)$$

then: $\|c_i\|_\infty = \|r_i\|_\infty$.

To reduce round-off error we use a $D$ of the form:

$$D = \mathrm{diag}(b^{i_1}, b^{i_2}, \ldots, b^{i_n})$$

where $b$ is the radix of the computer (usually $b=2$).

<u>Algorithm</u> : (Osborne balancing).

$D = \mathrm{diag}(1,1,\dots,1)$. , $b = 2$

do

  loop $i = 1,\dots,n$

  $c_i = \|i^{th}$ column of $A\|_\infty$

  $r_i = \|i^{th}$ row of $A\|_\infty$

$$k_i = \begin{cases} 0 & , \text{ if } bc_i \geq r_i \\ \max\{m \in \mathbb{N} : c_i b^{2m+1} < r_i\} & , \text{ if otherwise} \end{cases}$$

$$l_i = \begin{cases} 0 & , \text{ if } c_i b^{2k_i - 1} \leq r_i \\ \max\{m \in \mathbb{N} : c_i b^{2k_i - 2m - 1} > r_i\} & , \text{ if otherwhise} \end{cases}$$

  $d_i = b^{k_i - l_i}$

  endloop

  $P = \mathrm{diag}(d_1, d_2, \dots, d_n)$.

  $A = P^{-1} A P$.

  $D = DP$

while (terminating condition = false)

where the terminating condition is

$$\forall i \in [n] : \frac{c_i d_i^2 + r_i}{d_i} \leq 0.95 \cdot (c_i + r_i).$$

<u>Thm</u> : This algorithm balances the matrix and terminates in finite time

## ▼ Hessenberg reduction

Def : Let $A \in \mathbb{C}^{n\times n}$. We say that $A$ is a Hessenberg matrix $\Leftrightarrow$ $\exists\, U, S \in \mathbb{C}^{n\times n}$ such that

a) $A = U + S$

b) $U$ upper triangular

c) $S = [s_{ij}]$ with $s_{ij} = \begin{cases} a_j & , \text{ if } i = j+1 \\ 0 & , \text{ otherwise.} \end{cases}$

example

The matrix

$$A = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ a_1 & u_{22} & u_{23} & u_{24} \\ 0 & a_2 & u_{33} & u_{34} \\ 0 & 0 & a_3 & u_{44} \end{bmatrix}$$

is a Hessenberg matrix.

The advantages of a Hessenberg matrix are:

a) Easy to compute $\det A \sim O(n^2)$.

b) Easy to compute QR decomposition $\sim O(n^2)$

c) Closed under a QR iteration

These are established by the following theorems:

Thm : Let $A \in \mathbb{C}^{n\times n}$ be a Hessenberg matrix and $A_k = [a_{ij}] \in \mathbb{C}^{k\times k}$ with $k \in [n]$ be the $n$ submatrices $A_1, A_2, \ldots, A_k$.

Then:

$$\det A_k = u_{kk} \det A_{k-1} + \sum_{m=1}^{k-1} \left[ (-1)^{k-m} u_{mk} \det A_{m-1} \prod_{j=m}^{k-1} a_j \right]$$

$$\det A_0 \equiv 1.$$

Thm : Let $A \in \mathbb{C}^{n\times n}$ be a Hessenberg matrix with QR decomposition $A = QR$.

Then $RQ$ is also a Hessenberg matrix.

To do a Hessenberg reduction we apply a Householder algorithm as follows:

<u>Algorithm</u>

$V = I$

loop $i = 1, 2, \dots, n-2$:

  Compute householder $H$ such that $HA$ zeroes out $i^{th}$ column

  $A = HAH^H$

  $V = HI$ # optional

endloop.

$\rightsquigarrow$ The Hessenberg reduction is : $B = VAV^H$ or $A = V^H B V$
where $B$ is a unitary matrix.
If you want to compute eigenvectors then $V$ must be computed otherwise it is not required.

We have already shown an efficient way to compute $HA$.
In particular:

$$y = H(u)x \Rightarrow y_i = x_i - 2\frac{\langle u|x\rangle}{\langle u|u\rangle} u_i \quad \text{for } y, x \in \mathbb{C}^{n\times 1},\ u \in \mathbb{C}^n.$$

Now consider $y = x H^H(u)$ for $y, x \in \mathbb{C}^{1\times n}$.

$$y = x H^H(u) = x\left(I - \frac{2}{\|u\|^2}(u \otimes u)\right)^H = x\left(I - \frac{2}{\|u\|^2}(u^* \otimes u^*)\right) =$$

$$= x - \frac{2}{\|u\|^2} x u^* u^{*H} = x - \frac{2}{\|u\|^2}\langle u|x\rangle u^{*H} \Rightarrow$$

$$\Rightarrow y_i = x_i - 2\frac{\langle u|x\rangle}{\langle u|u\rangle} u_i^*.$$

To compute $HA$ apply $y_i = x_i - 2\frac{\langle u|x\rangle}{\langle u|u\rangle} u_i$ to all <u>columns.</u>

To compute $AH^H$ apply $y_i = x_i - 2\frac{\langle u|x\rangle}{\langle u|u\rangle} u_i^*$ to all <u>rows.</u>

~~In general~~

To eliminate the $k$ column to Hessenberg form:
set $x = (\underbrace{0, 0, \dots, 0}_{k}, a_{k+1,k}, a_{k+2,k}, \dots, a_{nk})$

$z = (\underbrace{0, 0, \dots, 0}_{k}, 1, 0, \dots, 0)$ and $u = x \pm \|x\| z$

▼ Schur decomposition

The simplest way to compute the Schur decomposition is ~~as~~:

$H_0 = A$ , $P_0 = I$
loop $k = 1, 2, \dots$
$H_{k-1} = Q_k R_k$
$H_k = R_k Q_k$.
$P_k = P_{k-1} Q_k$
endloop.

If $A = P^H T P$ is the Schur decomposition of $A$, then
$\lim_{k \to +\infty} P_k = P$ and $\lim_{k \to +\infty} H_k = T$.

Note that each iteration (QR iteration) is a similarity transformation:

$$H_k = R_k Q_k = Q_k^H Q_k R_k Q_k = Q_k^H H_{k-1} Q_k. = Q_k^{-1} H_{k-1} Q_k.$$

Each QR decomposition is an $O(n^3)$ operation.
For a Hessenberg matrix, QR can be computed faster with Givens rotations. The algorithm used then is:

Algorithm

$H_0 = U^H A U$ , $P_0 = U$
loop $k = 1, 2, \dots$
$H_{k-1} = Q_k R_k$ $\rightsquigarrow$ using $n-1$ Givens rotations.
$H_k = R_k Q_k$
$P_k = P_{k-1} Q_k$.
endloop.

As a terminating condition monitor the change in the diagonal elements of $H_k$.

The eigenvalues of $A$ are the diagonal elements of $T$ where $A = P^H T P$.

<u>Thm</u>: If $A \in \mathbb{C}^{n\times n}$ is ~~symmetric~~ hermitian with Schur decomposition $A = P^H T P \Rightarrow T$ is diagonal. $T = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$

<u>Corollary</u>: $P = [v_1 v_2 \cdots v_n]$ are the eigenvectors of the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$.

In the general case, computing the eigenvectors is more difficult.

## <u>Finding eigenvectors with SVD.</u>

Let $A \in \mathbb{C}^{n\times n}$ and $\lambda \in \lambda(A)$. The eigenspace $E(\lambda; A)$ is:

$$E(\lambda; A) = \{x \in \mathbb{C}^n : Ax = \lambda x\} = \{x \in \mathbb{C}^n : (A - \lambda I)x = 0\}$$
$$= \mathrm{null}(A - \lambda I).$$

Let $U^H A V = D$ be the SVD decomposition of $A$ with $D = \mathrm{diag}\{\sigma_1, \sigma_2, \dots, \sigma_n\}$.
$V = [v_1 v_2 \cdots v_n]$.

~~If the matrix is not defective: $E(\lambda; A) =$~~

Usually $\dim E(\lambda; A) = 1 \Rightarrow E(\lambda; A) = \mathrm{span}\{v_n\}$.

In general, if $\sigma_{n-1} > 0 \wedge \sigma_n = 0$ then the eigenvector is just $v_n$.

If $\sigma_{k+1} = \dots = \sigma_n = 0$ then $E(\lambda; A) = \mathrm{span}\{v_{k+1}, \dots, v_n\}$.

This gives the <u>geometric</u> multiplicity of $\lambda$: $|\{v_{k+1}, \dots, v_n\}|$.

Comparing with the algebraic multiplicity we can detect defective matrices. This is the most robust way to deal with eigenvectors. To apply it we need a way to compute SVD.

Algorithm: (SVD decomposition)

Let $A \in \mathbb{C}^{n\times n}$.

1) Apply ~~gore~~ Givens rotations : $B = U_B^H A V_B$ to bring $A$ into bidiagonal form:

$$B = \begin{bmatrix} d_1 & f_1 & 0 & 0 \\ 0 & d_2 & f_2 & 0 \\ 0 & 0 & d_{n-1} & f_{n-1} \\ 0 & 0 & 0 & d_n \end{bmatrix}$$

2) Apply Golub-Kahan iterations on $B$ until it converges to diagonal form.

- The Golub-Kahan iteration

Let $B = \text{bidiag}(d_1, d_2, \dots, d_n; f_1, f_2, \dots, f_{n-1})$.

a) Set

$$T = \begin{bmatrix} d_m^2 + f_m^2 & d_m f_n \\ d_m f_n & d_n^2 + f_n^2 \end{bmatrix} \qquad m = n-1.$$

Let $\lambda_1, \lambda_2 \in \mathbb{C}$ be the eigenvalues of $T$. Set $\lambda$ equal to the one that ~~$\lambda = \min\{\lambda$~~ is closer to $d_n^2 + f_m^2$.

b) Find a givens rotation $G$ such that

$$G \begin{bmatrix} d_1^2 - \lambda \\ d_1 f_1 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}$$

c) Set $B = BG$. Now there is an unwanted nonzero element:

$$BG = \begin{bmatrix} d_1 & f_1 & 0 & 0 \\ + & d_2 & f_2 & 0 \\ 0 & 0 & d_{n-1} & f_{n-1} \\ 0 & 0 & 0 & d_n \end{bmatrix}$$

d) Use Givens rotations to chase $+$ down the diagonal.

$$B = U_1^H B = \begin{bmatrix} \times & \times & + & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}, \quad B = BV_2 = \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & + & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}$$

## Characteristic polynomial method

Method for
1) Computing the characteristic polynomial
2) Computing the eigenvectors if matrix not Hermitian.

**Thm** : Let $H \in \mathbb{C}^{n\times n}$ be a Hessenberg matrix of the form

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{1,n-1} & h_{1n} \\ a_1 & h_{22} & h_{2,n-1} & h_{2n} \\ 0 & a_2 & h_{n-1,n-1} & h_{n-1,n} \\ 0 & 0 & a_{n-1} & h_{nn} \end{bmatrix}$$

and let $\varphi_0(\lambda) = 1$ , $\varphi_1(\lambda) = h_{11} - \lambda$ and

$$\varphi_k(\lambda) = (h_{kk} - \lambda)\,\varphi_{k-1}(\lambda) + \sum_{m=1}^{k-1} (-1)^{k-m} h_{mk}\, \varphi_{m-1}(\lambda) \prod_{l=m}^{k-1} a_l$$

Then:

a) $\det(H - \lambda I) = \varphi_n(\lambda)$.

b) If $H$ is diagonalizable and $\lambda \in \lambda(H)$ an eigenvalue, then $x = \mathrm{col}(x_1, x_2, \dots, x_n)$ with

$$x_n = \varphi_{n-1}(\lambda)$$
$$x_k = (-1)^{n-k}\, \varphi_{k-1}(\lambda) \prod_{l=k}^{n-1} a_l \quad , \forall k \in [n-1]$$

is the corresponding eigenvector.

**Remark** : For tridiagonal matrices $H \in \mathbb{C}^{n\times n}$ the recursion reduces to

$$\varphi_k(\lambda) = (h_{kk} - \lambda)\,\varphi_{k-1}(\lambda) - h_{k-1,k}\, a_{k-1}\, \varphi_{k-2}(\lambda).$$

**Remark** : To find the eigenvectors of an arbitrary $A \in \mathbb{C}^{n\times n}$

a) Reduce to Hessenberg form : $A = U^H H U$

b) Compute eigenvectors of $H$. Then the diagonalization theorem says that $H = X^{-1} D X$ where $D = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, where $X = [v_1 v_2 \cdots v_n]$ are the eigenvectors of $H$.

$$B = U_2^H B = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & + \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}, \quad B = BV_3 = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & + & x \end{bmatrix}$$

$$B = U_3^H B = \begin{bmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}$$

Overall $B = (U_{n-1}^H \cdots U_1^H) B (G_1 V_2 \cdots V_{n-1})$.

<u>Remark</u>: To apply this iteration it is necessary for $f_i \neq 0$.
Typical termination condition is:

$$|f_i| \leq \varepsilon(|d_i| + |d_{i+1}|), \quad \forall i \in [n].$$

If for some $k \in [n]$ $f_k = 0$, then we must split

$$B = \begin{array}{cc} \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} & \begin{matrix} k \\ n-k \end{matrix} \\ \begin{matrix} k & n-k \end{matrix} & \end{array}$$

and apply further Golub-kahan iterations on $B_1, B_2$.
A direct way to do this is the following algorithm:

<u>Algorithm</u>

$B = U^H A V \rightsquigarrow$ bidiagonalize $A$

do until $q = n$.

loop $i = 1, \dots, n-1$ : if $|a_{i,i+1}| \leq \varepsilon(|a_{ii}| + |a_{i+1,i+1}|)$ then $a_{i,i+1} = 0$

find largest $q$ and smallest $p$ such that

$$B = \begin{array}{cc} \begin{bmatrix} B_{11} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{bmatrix} & \begin{matrix} p \\ n-p-q \\ q \end{matrix} \\ \begin{matrix} p & n-p-q & q \end{matrix} & \end{array}$$

$B_{33} \rightsquigarrow$ diagonal

$B_{22} \rightsquigarrow$ nonzero superdiagonal

if $q < n$

if any diagonal entry in $B_{22}$ is zero then zero superdiagonal entry on same row

else: Apply Golub-kahan iteration to $B_{22}$.

endif

Remark: To find the eigenvectors of an $A \in \mathbb{C}^{n \times n}$.

a) Reduce to Hessenberg form: $U_0^H A U_0 = H$.

b) Compute $\lambda(A)$ by Schur decomposition.

c) Compute eigenvectors of $H$: $z_1, z_2, \ldots, z_n \in \mathbb{C}^n$.

d) Obtain eigenvectors for $A$: $v_k = U_0 z_k$.